

UNITED STATES PATENT APPLICATION

Determining a Remote Device Name

INVENTOR

Roger Woodruff

Citizenship: **United States of America**

Residence: **Mound, MN**

Schwegman, Lundberg, Woessner, & Kluth, P.A.

1600 TCF Tower

121 South Eighth Street

Minneapolis, Minnesota 55402

ATTORNEY DOCKET 1370.018US1

CLIENT REF. 102628 / 3779

Determining a Remote Device Name

5

FIELD OF THE INVENTION

This invention relates generally to networks of computers and more particularly to accessing a device at a remote computer via a network.

BACKGROUND OF THE INVENTION

10

Computer systems need a way to store and retrieve data from a variety of data devices, such as disk drives, printers, display screens, and scanners. In the past, each computer typically had its own directly attached devices, which no other computer was capable of using. But, this was a cumbersome and expensive design because sharing data among computers was difficult and a particular device might stay idle and unused for lengthy periods. For example, a retail store might have multiple cash registers, and each cash register storing its own price/product data locally is less efficient and more cumbersome than every cash register accessing the same price/product data on one common disk drive. Also, a printer is a device that a computer user might need for only short periods, so each computer attaching its own local printer is more expensive than multiple computers sharing one printer.

15

20

25

In order to overcome the aforementioned cumbersome and expensive solutions, computers were connected in networks, and one computer was allowed to store and retrieve data from another computer's data devices. But, accessing data devices on another computer created the problem of how to know what devices this other computer had available. Previous systems required the user to manually determine the identifiers (names) of the available devices attached to other computers

on the network and enter these identifiers into the user's own computer. This manual process is inefficient, error prone, and annoying for the user. What is needed is a solution that allows one computer to automatically determine the identifiers of devices attached to remote computers.

5

SUMMARY OF THE INVENTION

The present invention provides solutions to the above-described shortcomings in conventional approaches, as well as other advantages apparent from the description below. A method, system, and signal-bearing media are provided for determining the name of a remote device. In one aspect, a server discovers the devices attached to it and extracts the device names in a first protocol format. The server encodes the device names into a second protocol format. When a client requests a list of supported devices, the server sends the device names found in the second protocol format. The client decodes the names from the second protocol format back into the first protocol format and presents the device names to a host attached to the client. In this way, the host is freed from manually predetermining the device names, and the host need have no knowledge of the server or the second protocol.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 depicts a pictorial example of a network of computer systems that can be used to implement an embodiment of the invention.

Fig. 2A depicts a block diagram of some of the principal components of a computer system that can be used to implement an embodiment of the invention.

25

Fig. 2B depicts a block diagram of some of the principal components of a computer system that can be used to implement an embodiment of the invention.

Fig. 3 depicts an example flowchart that describes the operation of an embodiment of the invention.

5 Fig. 4 depicts an example flowchart that describes the operation of an embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

10 In the following detailed description of exemplary embodiments of the invention, reference is made to the accompanying drawings (where like numbers represent like elements), which form a part hereof, and in which is shown by way of illustration specific exemplary embodiments in which the invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention, but other embodiments may be utilized and logical, mechanical, electrical, and other changes may be made without departing from the scope of the present invention. The following detailed description is, 15 therefore, not to be taken in a limiting sense, and the scope of the present invention is defined only by the appended claims.

20 Fig. 1 depicts example system 100, including a network of computer systems and devices that can be used to implement an embodiment of the invention. Host computer 105 is communicatively coupled to channel fabric 110, which is communicatively coupled to client computer 115, which is communicatively coupled to network 120. Network 120 is communicatively coupled to server computer 125, which is communicatively coupled to channel fabric 130, which is communicatively coupled to device 135.

25 Host 105 is a computer that wishes to store and/or retrieve data to/from device 135. In one embodiment, host 105 interfaces to channel fabric 110 as if host

105 were directly attached to device 135, so that host 105 has no knowledge of the existence of client 115, network 120, server 125, or channel fabric 130.

Channel fabric 110 transfers data between host 105 and client 115. Channel fabric 130 transfers data between server 125 and device 135. In one embodiment, channel fabric 110 and 130 are implemented using the Fibre Channel I/O (Input/Output) protocol. Fibre Channel is an asynchronous, serial I/O protocol that is unaware of the content or meaning of the information being transferred. In other embodiments, any suitable I/O protocol can be used, such as the SCSI (Small Computer System Interface) or IDE (Integrated Device Electronics) protocols.

Networked systems often follow a client/server architecture. In one embodiment, a client is a process (i.e., a program or task) executing on a computer that requests a service provided by another program or computer. In another embodiment, the client is the computer itself. The client utilizes the requested service without needing to know any working details about the other program or the service itself. In networked systems, a client is usually a computer that accesses shared network resources provided by another computer, such as server.

Client 115 decodes the name of device 135, which client 115 retrieves across network 120 from server 125 and presents the decoded name to host 105 via channel fabric 110. In one embodiment, client 115 emulates the interface of device 135, meaning that client 115 appears to host 105 as if client 115 were actually device 135. Client 115 is further described below with reference to Fig. 2A.

Network 120 can include many servers and/or many clients, which act to pass information between them. In one embodiment, network 120 utilizes the TCP/IP protocol. TCP/IP is an acronym for "Transport Control Protocol/Internet Protocol," a protocol developed by the Department of Defense for facilitating communications

between computers. In one embodiment, iSCSI protocol commands are sent through network 120 using TCP/IP. iSCSI is a protocol for sending SCSI (Small Computer System Interface) commands over the Internet. SCSI is a standard high-speed parallel interface defined by the X3T9.2 committee of the American National Standards Institute (ANSI).

Server 125 is a computer remote from client 115 over the network 120. Based on requests from client 115, server 125 scans and searches for information sources and presents filtered, electronic information to client 115 as server responses. Server 125 is thus a network computer that runs administrative software that controls access to all or part of a network and its resources, such as data on device 135. Server 125 is further described below with reference to Fig. 2B.

Device 135 can be any I/O device, such as a workstation, hard disk drive, disk array, diskette drive, CD-ROM drive, DVD (Digital Video Disc) drive, tape drive, scanner, medical instrument, or any other device capable of receiving and/or sending data. Device 135 contains device name 140, which uniquely identifies device 135.

In one embodiment, device name 140 is a World Wide Name (WWN). World Wide Names can be two types: World Wide Node Names (WWNN) and World Wide Port Names (WWPN). A node can have only one node name, but each port it supports will have a port name. Each attachment has an associated node and port name. The port name is typically used to uniquely identify the device and the path to the device at the same time.

In another embodiment, device name 140 is a target number/logical unit number, which identifies a device on the SCSI Parallel Interface. This is the first

level of device identification. In addition, each Logical Unit on a target may have a serial number, or even a World Wide Name.

5 In still another embodiment, device name 140 is a serial number, which is used by the Fibre Channel protocol.

10 In one embodiment of system 100, a user or software application at host 105 issues a command to store or retrieve data on device 135. The request is processed by host 105 into a Fibre Channel command and sent over channel fabric 110 to client 115, who converts the Fibre Channel command into one or more SCSI commands. Client 115 then encapsulates the commands and data by representing them as a serial string of bytes proceeded by iSCSI headers. Client 115 then uses a TCP/IP layer to break the encapsulated data into packets suitable for transfer over the network according to the TCP/IP protocol.

15 Client 115 then sends the packets over network 120. Server 125 recombines the packets into the original encapsulated SCSI commands and data. Server 125 then converts the SCSI commands and data into Fibre Channel commands, and sends them across channel fabric 130 to device 135, which performs the functions that were originally requested by host 105. If a request for data has been sent, the data is retrieved from the drive, encapsulated and returned to the requesting computer.

25 Although system 100 has been described in the context of TCP/IP, Fibre Channel, and iSCSI, any suitable protocols can be used. For example, SCSI or IDE can be used instead of Fibre Channel for one or both of the interfaces between client 115 and host 105 and between server 125 and device 135.

Fig. 2A depicts a block diagram of the principal components of client 115 attached to network 120 and host 105. Client 115 contains memory 230 connected via bus 255 to storage 235, processor 240, channel adapter 245, and network adapter 250. Although the various components of Fig. 2A are drawn as single entities, each may consist of multiple entities and may exist at multiple levels.

Memory 230 comprises an number of individual, volatile-memory modules that store segments of operating system and application software while power is supplied to client 115. The software segments are partitioned into one or more virtual memory pages that each contain an uniform number of virtual memory addresses. When the execution of software requires more pages of virtual memory than can be stored within memory 230, pages that are not currently needed are swapped with the required pages, which are stored within non-volatile storage devices 122 or 123. Memory 230 is a type of memory designed such that the location of data stored in it is independent of the content. Also, any location in memory 230 can be accessed directly without needing to start from the beginning.

Memory 230 contains decoder 260, which contains instructions capable of being executed by processor 240. In another embodiment, decoder 260 can be implemented by control circuitry though the use of logic gates, programmable logic devices, or other hardware components in lieu of a processor-based system. Although decoder 260 is shown contained within memory 230, in another embodiment, decoder 260 is a part of channel adapter 245. Decoder 260 decodes device name 140 and presents it to host 105. The operations of decoder 260 are further described below with reference to Fig. 4.

Processor 240 executes instructions and includes that portion of client 115 that controls the operation of the entire computer system, including executing the arithmetical and logical functions contained in a particular computer program. Processor 240 organizes data and program storage in memory 230 and transfers data and other information between the various part of the computer system. Processor 240 accesses data and instructions from and stores data to memory 230.

Any appropriate processor can be utilized to implement processor 240. Although client 115 is shown to contain only a single processor and a single system bus, the present invention applies equally to computer systems that have multiple processors and to computer systems that have multiple buses that each perform different functions in different ways.

To support storage and retrieval of data, client 115 further includes storage 235. In one embodiment, storage 235 is one or more hard disk drives. In another embodiment, storage 235 can be ROM (read only memory), a tape drive, a diskette drive, a CD-ROM drive, or any device or combination of devices capable of storing instructions and data. Although storage 235 is shown incorporated into client 115, in other embodiments, it can be external to client 115, either connected directly, on a local area network (LAN), on network 120, or as part of device 135.

Client 115 includes network adapter 250, which facilitates communication between client 115 and network 120, which might be a local area network (LAN), an intranet, or the Internet. Network adapter 250 can also be a modem, which supports communication between client 115 and another computer system over a standard telephone line. Furthermore, through a modem, client 115 can access other sources such as server, an electronic bulletin board, and the Internet or World Wide Web.

Network 120 provides a user of client 115 with a means of electronically communicating information, including software, with a remote computer or a network logical-storage device. In addition, network 120 can support distributed processing, which enables client 115 to share a task with other computer systems linked to the network. Network 120 may include a plurality of networks, each of which could include a plurality of individual computers. Network 120 and server 125 could be located a great geographic distance from client 115, or they could be in the same room or even on the same desktop. Client 115 can be connected to network 120 via a standard telephone line, a dedicated cable, or a wireless communications link.

Client 115 can be implemented using any suitable computer such as a Cisco SN5420 Universal Access Server. Portable computers, laptop computers, and network computers or Internet appliances are other possible configurations. The hardware depicted in Fig. 2A may vary for specific applications. For example, other peripheral devices such as optical-disk media, audio adapters, or chip programming devices, such as PAL or EPROM programming devices may be used in addition to or in place of the hardware already depicted. Thus, an embodiment of the invention can apply to any hardware configuration that allows attachment of devices, regardless of whether the hardware configuration is a complicated, multi-user computing apparatus, a single-user workstation, or a network appliance that does not have non-volatile storage of its own.

Referring to Fig. 2B, server 125 contains memory 270, network adapter 274, processor 275, storage 278 and channel adapter 280, which are all connected via system bus 290. Server 125 is capable of communicating across network 120 using a

TCP/IP (Transmission Control Protocol/Internet Protocol) connection, although any suitable communications protocol could be used.

Memory 270 can be any type of computer memory, analogous to that
5 described for memory 230. Memory 270 includes encoder 295, which contains instructions capable of being executed by processor 275. In another embodiment, encoder 295 could be implemented by control circuitry through the use of logic gates, programmable logic devices, or other hardware components in lieu of a processor-based system. Although encoder 295 is shown contained within memory 270, in
10 another embodiment, encoder 295 is a part of channel adapter 280. Encoder 295 is further described below with reference to Fig. 3.

Processor 275 can be any type of computer processor, analogous to those described for processor 240. Processor 275 accesses data and instructions from and
15 stores data to storage 278. Storage 278 can be any type of non-volatile storage, analogous to that described for storage 235.

Server 125 can be implemented using any suitable computer such as a Cisco SN5420 Universal Access Server. Portable computers, laptop computers, and
20 network computers or Internet appliances are other possible configurations. The hardware depicted in Fig. 2B may vary for specific applications. For example, other peripheral devices such as optical-disk media, audio adapters, or chip programming devices, such as PAL or EPROM programming devices may be used in addition to or in place of the hardware already depicted. Thus, an embodiment of the invention can
25 apply to any hardware configuration that allows attachment of devices, regardless of whether the hardware configuration is a complicated, multi-user computing

apparatus, a single-user workstation, or a network appliance that does not have non-volatile storage of its own.

As will be described in detail below, aspects of an embodiment pertain to specific method steps implementable on computers. In another embodiment, the invention can be implemented as a computer program product for use with a computer system. The programs defining the functions of the embodiment can be delivered to a computer via a variety of signal-bearing media, which include, but are not limited to:

(1) information permanently stored on non-rewriteable storage media (e.g., read only memory devices within a computer such as CD-ROM disks) readable by a CD-ROM drive;

(2) alterable information stored on writeable storage media (e.g., a hard disk drive or diskette); or

(3) information conveyed to a computer by a communications media, such as through a computer or telephone network including wireless communications.

Such signal-bearing media, when carrying computer-readable instructions that direct the functions of the present invention, represent embodiments of the present invention.

Fig. 3 depicts an example flowchart that describes the operation of an embodiment of the invention at server 125. Control begins at block 300. Control then continues to block 310 where network adapter 274 discovers device 135 attached via channel fabric 130 and passes device name 140 to encoder 295. Control then continues to block 320 where encoder 295 encodes device name 140 into a target acquired name. In one embodiment, the target acquired name is in the iSCSI format. For example, if device name 140 is represented in hexadecimal notation is

0x2200000001020304, the ASCII iSCSI target acquired name can be
“disk/WWPN/22:00:00:00:01:02:03:04”. “Disk” identifies the target acquired name
and “WWPN” represents the device identifier type of World Wide Port Name; other
types can be “SCSI” or “SERNO” (Serial Number). The character “/” separates the
5 various fields and the character “:” separates bytes of binary data. In other
embodiments, any separation characters or other appropriate delimiters can be used.
Control then continues to block 330 where network adapter 274 detects that the client
has asked for the target acquired name and sends the encoded target acquired name to
client. Control then continues to block 399 where the function returns.

10 Fig. 4 depicts an example flowchart that describes the operation of an
embodiment of the invention at client 115. Control begins at block 400. Control
then continues to block 410 where client 115 requests server 125 to send target
acquired names for devices attached to server 125. Control then continues to block
15 420 where decoder 260 searches for encoded device names in the target acquired
names returned from server 125. Controller then continues to block 430 where
decoder 260 decodes the target acquired names into device names and presents them
to host 105. Control then continues to block 499 where the function returns.